**Assignment 12.3**

**Explain in detail.**

**●What is meant by FlumeNG?**

FlumeNG is a real time loader for streaming your data into Hadoop. It stores data in HDFS and HBase. To solve certain known issues and limitations, Flume requires a refactoring of some core classes and systems. This bug is a parent issue to track the development of a "Flume NG.

Here is a summary of concepts that Flume NG introduces, redefines, or reuses from earlier implementation:

Event: A byte payload with optional string headers that represent the unit of data that Flume can transport from it’s point of origination to it’s final destination.

Flow: Movement of events from the point of origin to their final destination is considered a data flow, or simply flow. This is not a rigorous definition and is used only at a high level for description purposes.

Client: An interface implementation that operates at the point of origin of events and delivers them to a Flume agent. Clients typically operate in the process space of the application they are consuming data from. For example, Flume Log4j Appender is a client.

Agent: An independent process that hosts flume components such as sources, channels and sinks, and thus has the ability to receive, store and forward events to their next-hop destination.

Source: An interface implementation that can consume events delivered to it via a specific mechanism. For example, an Avro source is a source implementation that can be used to receive Avro events from clients or other agents in the flow. When a source receives an event, it hands it over to one or more channels.

Channel: A transient store for events, where events are delivered to the channel via sources operating within the agent. An event put in a channel stays in that channel until a sink removes it for further transport. An example of channel is the JDBC channel that uses a file-system backed embedded database to persist the events until they are removed by a sink. Channels play an important role in ensuring durability of the flows.

Sink: An interface implementation that can remove events from a channel and transmit them to the next agent in the flow, or to the event’s final destination. Sinks that transmit the event to it’s final destination are also known as terminal sinks. The Flume HDFS sink is an example of a terminal sink. Whereas the Flume Avro sink is an example of a regular sink that can transmit messages to other agents that are running an Avro source.

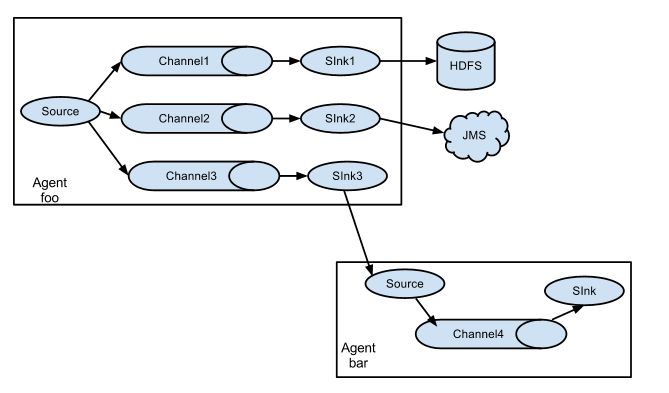
**● Can Flume provides 100 % reliability to the data flow?**

Yes, it provides end-to-end reliability of the flow. By default flume uses a transactional approach in the data flow. Sources and sink encapsulated in a transactional repository provides by the channels. This channels responsible to pass reliably from end to end in the flow. So it provides 100% reliability to the data flow.

**● Can Flume distribute data to multiple destinations?**

Yes, it supports multiplexing flow. The event flows from one source to multiple channels and multiple destinations. It’s achieved by defining a flow multiplexer.

In above example, data flows and replicated to HDFS and another sink to destination and another destination is input to another agent.



**● Explain about the different channel types in Flume. And which channel type is faster?**

The 3 different built in channel types available in Flume are-

MEMORY Channel – Events are read from the source into memory and passed to the sink.

JDBC Channel – JDBC Channel stores the events in an embedded Derby database.

FILE Channel –File Channel writes the contents to a file on the file system after reading the event from a source. The file is deleted only after the contents are successfully delivered to the sink.

MEMORY Channel is the fastest channel among the three however has the risk of data loss. The channel that you choose completely depends on the nature of the big data application and the value of each event.